

**SKRIPSI**

**IMPLEMENTASI ALGORITMA FAST FOURIER  
TRANSFORM UNTUK PENGOLAHAN SINYAL DIGITAL  
PADA TUNING GITAR DENGAN OPEN STRING**



**Oleh:**

**ROBBY DIANPUTRA**

**G1A010007**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS BENGKULU  
2014**

**IMPLEMENTASI ALGORITMA FAST FOURIER  
TRANSFORM UNTUK PENGOLAHAN SINYAL DIGITAL  
PADA TUNING GITAR DENGAN OPEN STRING**

**SKRIPSI**

**Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana pada  
Fakultas Teknik Program Studi Teknik Informatika  
Universitas Bengkulu**



**OLEH  
ROBBY DIANPUTRA  
G1A010007**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS BENGKULU  
2014**

## MOTTO DAN PERSEMBAHAN

### Motto :

*“Jika Kalian Menyerah, maka Habislah Sudah”*

### Persembahan :

System.out.println(“

- Allah SWT
- Muhammad SAW
- Mak Dan Bak Tersayang Dan Tercinta Yang Selalu Mendoakan Dan Mendorong Semangat Belajar Demi Masa Depan Anaknya Yang Lebih Cerah.
- Wah Meswarni Agustin , Enga Ike Noviarti Dan Adekku Vebby Okta Syaputra Yang Selalu Memberi Semangat Dan Motivasi.
- Seorang Yang Tersayang Adinda Paramita Intan Nur Safitri, Yang Memberi Ide Dan Semangat Serta Menjadi Tempat Curhat.
- Teman – Teman Anggota Gank Ganteng : Fauzi , Deki Satria Dan Muhammad Adi Baskoro , Yang Selalu Menemani Dari Awal Hingga Lulus Ini.
- Sahabat – Sahabat Angkatan 2010 Yang Tercinta..
- Universitas Bengkulu , Fakultas Teknik Dan Prodi Teknik Informatika.

“);

## KATA PENGANTAR



*Assalamu'alaikum Wr.Wb.*

Alhamdulillahirabbil'alamin, puji syukur penulis ucapkan kepada Allah SWT yang telah memberikan rahmat serta hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Implementasi algoritma *fast fourier transfrom* untuk pengolahan sinyal digital pada *tuning* gitar”.Penulisan skripsi ini disusun sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik di Program Studi Teknik Informatika Universitas Bengkulu.

Selesainya penyusunan skripsi ini tidak terlepas dari bimbingan, arahan, masukan dan bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak Khairul Amri, S.T., M.T sebagai Dekan Fakultas Teknik Universitas Bengkulu.
2. Ibu Desi Andreswari, S.T., M.Cs sebagai Ketua Program Studi Teknik Informatika.
3. Ibu Dr. Diah Puspitaningrum ,S.T., M.Kom sebagai dosen pembimbing utama yang telah meluangkan waktu memberikan bimbingan dan motivasi dalam menyelesaikan skripsi ini.
4. Ibu Ernawati ,S.T., M.Cs sebagai dosen pembimbing pendamping yang telah meluangkan waktu memberikan bimbingan dan motivasi dalam menyelesaikan skripsi ini.

5. Bapak Rusdi Efendi ,S.T., M.Kom sebagai dosen penguji utama dan Bapak Funny Farady C, S.Kom., M.T sebagai dosen penguji pendamping yang telah memberikan masukan-masukan demi penyempurnaan skripsi ini.
6. Seluruh dosen dan staf karyawan, serta seluruh civitas akademika Program Studi Teknik Informatika Fakultas Teknik Universitas Bengkulu.
7. mak, bak, wah, enga dan adek yang telah mendoakan, memberikan dukungan dan memotivasi dalam menyelesaikan skripsi ini.
8. Seseorang yang spesial yang telah mendoakan, memberikan dukungan dan menginspirasi judul skripsi ini, hingga memotivasi dalam menyelesaikan skripsi ini.
9. Teman-teman seperjuangan Teknik Informatika Angkatan 2010.
10. Semua pihak yang sudah membantu dan tidak dapat penulis sebutkan satu persatu.

Penulis berharap semoga Allah SWT mengaruniakan rahmat dan hidayah-Nya kepada mereka semua. Semoga skripsi ini dapat bermanfaat bagi kita semua, *aamiin*.

*Wassalamu'alaikum Wr. Wb.*

Bengkulu, Juni 2014

Penulis

# **Implementasi algoritma Fast Fourier Transform Untuk Pengolahan Sinyal Digital Pada Tuning Gitar dengan Open String**

Oleh  
Robby Dianputra  
NPM. G1A010007  
E-mail : rdp10007@gmail.com

## **ABSTRAK**

Penelitian ini bertujuan untuk mengetahui tingkat akurasi dalam pengolahan sinyal digital pada *tuning* gitar dengan *open string*. Dalam penelitian ini, algoritma yang digunakan adalah algoritma *Fast Fourier Transform* (FFT). Algoritma *Fast Fourier Transform* adalah algoritma untuk menghitung nilai *Discrete Fourier Transform* (DFT). DFT adalah metode untuk melakukan transformasi pada domain waktu menjadi domain frekuensi. Langkah-langkah penelitian meliputi analisis kebutuhan sistem, pengumpulan data, perancangan perangkat lunak dan pengujian sistem. Pada penelitian ini, sebagai masukan adalah suara gitar yang dipetik secara *open string* yaitu tanpa kunci, sedangkan sebagai keluaran adalah nilai frekuensi senar gitar dan pencocokan nada gitar. Perancangan perangkat lunak ini menggunakan bahasa pemrograman Matlab R2008b. Pengujian dilakukan untuk mencari tahu nilai tingkat akurasi FFT pada *tuning* gitar listrik dan bukan listrik. Selain itu, pengujian dilakukan untuk mengetahui kemampuan sistem melakukan pencocokan nada gitar. Dari penelitian yang telah dilakukan, FFT mampu mendapatkan tingkat akurasi terbaik yaitu 99.43%

Kata kunci: *Fast Fourier Transform*, *Tuning Gitar*, Pengolahan sinyal digital.

# **Implementation of FastFourier TransformAlgorithm for Digital Signal Processing in Guitar Tuning with Open String**

By  
ROBBY DIANPUTRA  
NPM. G1A010007  
E-mail : rdp10007@gmail.com

## **ABSTRACT**

The purpose of this research is to know the accuration rate of digital signal processing in guitar tuning with open string. This research using Fast Fourier Transform (FFT) Algorithm. Fast Fourier Transform (FFT) Algorithm is algorithm which is used to calculate values of discrete fourier transform (DFT). DFT is a method to transform time domain into frequency domain. Steps in this research are including analysis system requirements, data collection, design and testing of software systems. In this research, the input is open string guitar sound , while the output is the value of the frequency of a guitar string and matching guitar tone. This application is is built using Matlab R2008b. Several tests are conducted to find out the value of FFT accuracies on guitar tuning either in electric or non electric guitar. In addition, testing was conducted to determine the ability of the system to match the tone of the guitar. From the research, the FFT score for the best accuracy rate is 99.43%

*Keywords : Fast Fourier Transform, Tuning Gitar, Digital Signal Processing.*

## DAFTAR ISI

|                                                 |                                     |
|-------------------------------------------------|-------------------------------------|
| HALAMAN JUDUL.....                              | i                                   |
| HALAMAN PERSETUJUAN .....                       | <b>Error! Bookmark not defined.</b> |
| LEMBAR PENGESAHAN SKRIPSI .....                 | <b>Error! Bookmark not defined.</b> |
| MOTTO DAN PERSEMBAHAN .....                     | iv                                  |
| KATA PENGANTAR .....                            | v                                   |
| ABSTRAK .....                                   | vii                                 |
| ABSTRACT .....                                  | viii                                |
| DAFTAR ISI.....                                 | ix                                  |
| DAFTAR GAMBAR .....                             | xi                                  |
| DAFTAR TABEL .....                              | xii                                 |
| DAFTAR LAMPIRAN .....                           | xiii                                |
| BAB I PENDAHULUAN .....                         | 1                                   |
| 1.1 Latar Belakang .....                        | 1                                   |
| 1.2 Perumusan Masalah.....                      | 3                                   |
| 1.3 Batasan Masalah.....                        | 4                                   |
| 1.4 Tujuan Penelitian.....                      | 4                                   |
| 1.5 Manfaat Penelitian.....                     | 4                                   |
| 1.6 Sistematika Penelitian .....                | 4                                   |
| BAB II LANDASAN TEORI .....                     | 6                                   |
| 2.1 Pengolahan Sinyal .....                     | 6                                   |
| 2.2 Pengertian Sinyal.....                      | 7                                   |
| 2.2.1 Konsep frekuensi.....                     | 8                                   |
| 2.3 Algoritma Fast Fourier Transform (FFT)..... | 9                                   |
| 2.3.1 FFT Decimation In Time .....              | 10                                  |
| 2.3.2 FFT Decimation in Frequency .....         | 11                                  |
| 2.3.3 Contoh Perhitungan DFT .....              | 12                                  |
| 2.3 Alat Musik Gitar.....                       | 14                                  |
| 2.4 Model Pengembangan <i>Waterfall</i> .....   | 16                                  |
| 2.5 Pemrograman Terstruktur .....               | 19                                  |
| 2.6 Data Flow Diagram (DFD) .....               | 20                                  |
| 2.6.1 Pembagian Data Flow Diagram (DFD) .....   | 20                                  |
| 2.6.2 Komponen Data Flow Diagram .....          | 21                                  |
| BAB III METODE PENELITIAN.....                  | 23                                  |

|                                                       |           |
|-------------------------------------------------------|-----------|
| 3.1 Jenis Penelitian .....                            | 23        |
| 3.2 Jenis dan Sumber Data .....                       | 23        |
| 3.3 Objek Penelitian .....                            | 24        |
| 3.4 Metode Pengumpulan Data .....                     | 25        |
| 3.5 Metode Pengembangan Sistem .....                  | 25        |
| 3.6 Sarana Pendukung .....                            | 27        |
| 3.6.1 Perangkat Lunak.....                            | 27        |
| 3.6.2 Perangkat Keras .....                           | 28        |
| 3.7 Metode Pengujian Sistem.....                      | 28        |
| 3.8 Jadwal Penelitian.....                            | 28        |
| <b>BAB IV ANALISIS DAN PERANCANGAN SISTEM .....</b>   | <b>30</b> |
| 4.1 Identifikasi Permasalahan .....                   | 30        |
| 4.2 Analisis Sistem .....                             | 31        |
| 4.2.1 Pemahaman Kerja Sistem yang ada.....            | 31        |
| 4.2.2 Analisis Fungsional.....                        | 33        |
| 4.3 Perancangan Sistem.....                           | 34        |
| 4.3.1 Perancangan <i>Data Flow Diagram</i> (DFD)..... | 35        |
| 4.3.2 Perancangan Antarmuka .....                     | 40        |
| <b>BAB V HASIL DAN PEMBAHASAN .....</b>               | <b>45</b> |
| 5.1 Implementasi Antar Muka.....                      | 45        |
| 5.2 Pengujian Sistem .....                            | 48        |
| 5.2.1 Pengujian White Box .....                       | 48        |
| 5.2.2 Pengujian Black Box.....                        | 55        |
| <b>BAB VI PENUTUP .....</b>                           | <b>62</b> |
| 6.1 Kesimpulan.....                                   | 62        |
| 6.2 Saran.....                                        | 62        |
| <b>DAFTAR PUSTAKA .....</b>                           | <b>64</b> |
| <b>LAMPIRAN.....</b>                                  | <b>65</b> |

## DAFTAR GAMBAR

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| Gambar 2.1 Proses Sampling Sinyal Analog Menjadi Sinyal Digital .....   | 7  |
| Gambar 2.2 Proses Pengubahan Sinyal Analog Menjadi Sinyal Digital ..... | 7  |
| Gambar 2.3 Contoh Sinyal Suara.....                                     | 8  |
| Gambar 2.4 Waktu Sampling.....                                          | 9  |
| Gambar 2.5 Perubahan Struktur DFT DIT Menjadi DFT DIF .....             | 12 |
| Gambar 2.6 Model <i>Waterfall</i> .....                                 | 17 |
| Gambar 2.7 Fase-Fase Model Waterfall .....                              | 17 |
| Gambar 2.8 Ilustrasi Pemrograman Terstruktur.....                       | 19 |
| Gambar 4.1 Diagram Alir Sistem .....                                    | 31 |
| Gambar 4.2 Diagram Alir Pada Citra Uji.....                             | 33 |
| Gambar 4.3 Digram Konteks Aplikasi Tuner Gitar .....                    | 35 |
| Gambar 4.4 Diagram Level 1 Aplikasi Tuner Gitar .....                   | 37 |
| Gambar 4.5 Diagram Level 2 Proses 1 Aplikasi Tuner Gitar .....          | 39 |
| Gambar 4.6 Struktur Antarmuka Aplikasi .....                            | 40 |
| Gambar 4.7 Menu Utama Aplikasi .....                                    | 41 |
| Gambar 4.8 Sub Menu File .....                                          | 41 |
| Gambar 4.9 Sub Menu Help.....                                           | 42 |
| Gambar 4.10 Form Tuning Gitar .....                                     | 42 |
| Gambar 4.11 Form Tentang Gitar.....                                     | 43 |
| Gambar 4.12 Form Tentang Aplikasi.....                                  | 43 |
| Gambar 4.13 Form Keluar Aplikasi.....                                   | 44 |
| Gambar 5.1 Home.Fig.....                                                | 46 |
| Gambar 5.2 Tentangaplikasi.Fig.....                                     | 46 |
| Gambar 5.3 Tentanggitar.Fig .....                                       | 47 |
| Gambar 5.4 Guitartuner.Fig .....                                        | 47 |
| Gambar 5.5 Isi Menu File Pada Beranda Aplikasi.....                     | 48 |
| Gambar 5.6 Isi Menu Help Pada Beranda Aplikasi .....                    | 49 |
| Gambar 5.7 Kotak Dialog Keluar Beranda Aplikasi .....                   | 50 |
| Gambar 5.8 Tuning Gitar .....                                           | 51 |
| Gambar 5.9 Tampilan Form <i>Exit</i> .....                              | 53 |
| Gambar 5.10 Hasil Pengujian Terhadap Algoritma FFT .....                | 56 |
| Gambar 5.11 Hasil Pengujian Terhadap Algoritma FFT .....                | 56 |

## DAFTAR TABEL

|                                                                      |    |
|----------------------------------------------------------------------|----|
| Tabel 2.1 Nilai $F(X)$ .....                                         | 13 |
| Tabel 2.2 Perhitungan DFT Sinyal Suara .....                         | 13 |
| Tabel 2.3 Perhitungan FFT Dari Hasil DFT .....                       | 14 |
| Tabel 2.4 Frekuensi Nada Gitar .....                                 | 16 |
| Tabel 2.5 Komponen <i>Data Flow Diagram</i> .....                    | 21 |
| Tabel 3.1 Jadwal Pelaksanaan Penelitian.....                         | 29 |
| Tabel 4.1 Aliran Data Entitas User .....                             | 36 |
| Tabel 5.1 Daftar M.File Dan <i>Figure</i> Aplikasi .....             | 45 |
| Tabel 5.2 Source Code Menu File Dan Menu Help.....                   | 49 |
| Tabel 5.3 Source Code Sub Menu Tuning Gitar.....                     | 51 |
| Tabel 5.4 Source Code Sub Menu Exit.....                             | 53 |
| Tabel 5.5 Source Code Sub Menu Tentang Aplikasi.....                 | 54 |
| Tabel 5.6 Source Code Sub Menu Tetang Gitar .....                    | 55 |
| Table 5.7 Frekuensi Uji FFT Dengan Jarak 10 Cm Dalam Satuan Hz.....  | 57 |
| Table 5.8 Frekuensi Uji FFT Dengan Jarak 20 Cm Dalam Satuan Hz.....  | 58 |
| Table 5.9 Frekuensi Uji FFT Dengan Jarak 30 Cm Dalam Satuan Hz.....  | 59 |
| Table 5.10 Pembanding Uji FFT Dengan Jarak 10, 20 Dan 30 Cm .....    | 60 |
| Table 5.10 Frekuensi Uji FFT Dengan Garpu Tala Dalam Satuan Hz ..... | 60 |

## DAFTAR LAMPIRAN

|                                                                 |     |
|-----------------------------------------------------------------|-----|
| Lampiran A-1 Tabel Frekuensi Pengujian pada Senar Pertama ..... | A-1 |
| Lampiran A-2 Tabel Frekuensi Pengujian pada Senar Kedua .....   | A-2 |
| Lampiran A-3 Tabel Frekuensi Pengujian pada Senar Ketiga.....   | A-3 |
| Lampiran A-4 Tabel Frekuensi Pengujian pada Senar Keempat ..... | A-4 |
| Lampiran A-5 Tabel Frekuensi Pengujian pada Senar Kelima.....   | A-5 |
| Lampiran A-6 Tabel Frekuensi Pengujian pada Senar Keenam .....  | A-6 |
| Lampiran A-7 Tabel Frekuensi Pengujian pada Garpu Tala .....    | A-7 |
| Lampiran B-1 Hasil Pengujian Kode-Kode Sistem .....             | B-1 |
| Lampiran C-1 Tabel Frekuensi Senar .....                        | C-1 |
| Lampiran D-1 Gambar Interface Menu Utama.....                   | D-1 |
| Lampiran D-2 Gambar Interface Submenu File.....                 | D-1 |
| Lampiran D-3 Gambar Interface Submenu Help .....                | D-1 |
| Lampiran D-4 Gambar Interface Tuning Gitar .....                | D-2 |
| Lampiran D-5 Gambar Interface Tentang Gitar.....                | D-2 |
| Lampiran D-6 Gambar Interface Tentang Aplikasi .....            | D-2 |
| Lampiran D-7 Gambar Interface Exit .....                        | D-3 |

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Alat musik merupakan suatu instrumen yang bertujuan untuk menghasilkan nada. Salah satu contoh alat musik yang terkenal yaitu gitar , Gitar merupakan alat musik berdawai yang dimainkan dengan cara dipetik, umumnya menggunakan jari. Alat musik ini apabila baru atau sering digunakan maka harus dilakukan proses kalibrasi atau dikenal dengan istilah tuning (penyetelan nada).

Nada merupakan bunyi yang teratur dan mempunyai frekuensi yang tertentu (Wiflihani, 2013). Frekuensi yang dimiliki oleh senar gitar, berbeda dengan senar lainnya karena frekuensi dipengaruhi oleh panjang senar dan diameter senar. Dengan perbedaan frekuensi yang dihasilkan oleh senar, kita dapat mengolah frekuensi tersebut. Pengolahan pada frekuensi sering dikatakan dengan istilah *digital signal processing*.

Pengolahan sinyal digital (*digital signal processing*) sekarang diaplikasikan pada banyak bidang seperti pengolahan suara , musik, biomedical, navigasi dan telekomunikasi, serta pengolahan video dan gambar. Untuk melakukan pengolahan sinyal pada frekuensi, terdapat beberapa algoritma yang dapat digunakan, diantaranya Fast Fourier Transform, Gortzel , Hamming Window, transformasi wavelet packet dan lain – lain.

Algoritma Fast Fourier Transform (FFT) adalah suatu algoritma untuk menghitung Discrete Fourier Transform (DFT) yang digunakan untuk

menghitung spektrum frekuensi sinyal dan FFT merupakan prosedur penghitungan DFT yang efisien sehingga akan mempercepat proses penghitungan DFT yang secara substansial dapat lebih menghemat waktu dari pada metoda yang konvensional (Nandra Pradipta : 2011). Algoritma fast fourier transfrom membagi frekuensi per priodenya, karena itu algoritma ini dapat berkerja dengan baik sehingga menghasilkan akurasi dengan cepat dan efisien.

Penelitian terkait dalam pengolahan sinyal digital pernah dilakukan oleh Nandra Pradipta (2011). Dalam penelitiannya , algoritma Fast Fourier Transfrom (FFT) digunakan untuk menghitung spektrum frekuensi sinyal yang telah dicuplik komputer sehingga akan mempercepat proses penghitungan transformasi fourier diskrit. Algoritma ini dirasa cukup baik dalam melakukan pengolahan sinyal digital.

Penelitian lain dalam pengolahan sinyal digital juga pernah dilakukan oleh Nyoman Putra Sastra (2008) yang melakukan penelitian pengolahan sinyal digital menggunakan Algoritma Gortzel. Algoritma Gortzel mendeteksi frekuensi dengan tingkat keberhasilan pendeteksian sangat tergantung pada jumlah cuplikan (N), semakin banyak jumlah cuplikan maka semakin akurat pendeteksian frekuensi tersebut. Akan tetapi semakin banyak jumlah N, maka proses iterasinya menjadi lebih kompleks, sehingga waktu yang diperlukan untuk melakukan pendeteksian semakin lama. Untuk itu kesalahan dalam menentukan jumlah N dapat mempengaruhi akurasi dalam pengolahan sinyal digital.

Penelitian dengan studi kasus pengolahan sinyal digital terhadap gitar juga pernah dilakukan oleh Tanda Selamat dan Kevin Angkasa (2013) menggunakan algoritma karplus strong dengan berbasis komputer. Algoritma karplus strong merupakan algoritma sintesis wavetable yang melakukan modifikasi sendiri. Algoritma Karplus Strong dikembangkan oleh Alexander Strong dan dianalisis oleh Kevin Karplus sebagai suatu model untuk instrument musik pukul atau dipetik. Algoritma ini mensimulasikan dampak yang bernada tajam melalui sinyal pita lebar seperti dalam bentuk kumpulan noise. Sinyal tersebut diumpan balik melalui suatu delay line dimana panjangnya bergantung pada frekuensi dari not yang diinginkan. Sinyal tertunda tersebut dikirimkan melalui suatu filter lowpass untuk menghaluskan semua frekuensi lainnya kecuali frekuensi dari nada yang diinginkan. Pada penelitian ini telah menghasilkan aplikasi yang dapat melakukan tuning gitar dengan cara menghasilkan suara yang menyerupai suara gitar.

Berdasarkan latar belakang diatas, peneliti ingin mengimplementasikan algoritma fast fourier transform pada proses tuning sehingga penulis memilih judul **“Implementasi algoritma Fast Fourier Transform Untuk Pengolahan Sinyal Digital Pada Tuning Gitar dengan Open String ”**.

## **1.2 Perumusan Masalah**

Berdasarkan latar belakang diatas dapat dirumuskan permasalahan pada penelitian ini adalah “Bagaimana Mengimplentasikan algoritma Fast Fourier Transform pada Tuning Gitar dengan Open String? “

### 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini sebagai berikut:

1. Input sistem berupa suara yang dihasilkan oleh senar gitar yang diambil menggunakan microphone.
2. Output sistem berupa nilai frekuensi dan kecocokan nada gitar .
3. Objek Penelitian adalah gitar akustik yang memiliki enam senar.
4. Uji coba sistem dilakukan pada lingkungan hening.
5. Pengujian dilakukan dengan kondisi awal semua senar gitar bernada rendah.

### 1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah Mengetahui tingkat akurasi Algoritma Fast Fourier Transform pada Tuning Gitar dengan Open String.

### 1.5 Manfaat Penelitian

Manfaat yang ingin dicapai dari penelitian ini adalah dapat mengetahui tingkat akurasi algoritma *fast fourier transform* pada pengolahan sinyal digital dengan tuning gitar serta dapat melakukan tuning gitar.

### 1.6 Sistematika Penelitian

Dalam penyusunan skripsi ini, sistematika penulisan dibagi menjadi enam bab sebagai berikut:

#### **BAB I PENDAHULUAN**

Bab ini berisi pembahasan masalah umum yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

## **BAB II LANDASAN TEORI**

Pada bab ini menjelaskan tentang teori-teori yang menjadi dasar berfikir untuk penulisan skripsi dan pembangunan sistem.

## **BAB III METODE PENELITIAN**

Bab ini memuat uraian tentang metode-metode yang digunakan, baik dalam penelitian maupun pengembangan sistem.

## **BAB IV ANALISIS DAN PERANCANGAN SISTEM**

Dalam bab ini peneliti menguraikan analisis dan desain untuk merancang dan membangun aplikasi pengolahan sinyal digital pada tuning gitar dengan *open string* menggunakan algoritma *Fast Fourier Transform*.

## **BAB V HASIL DAN PEMBAHASAN**

Bab ini akan menguraikan pengkodean dan pengujian sistem yang dikembangkan dari tahapan analisis dan desain.

## **BAB VI KESIMPULAN DAN SARAN**

Bab ini berisi mengenai kesimpulan yang didapatkan dari rancang bangun sistem yang dibuat pada skripsi ini serta saran untuk pengembangan selanjutnya.

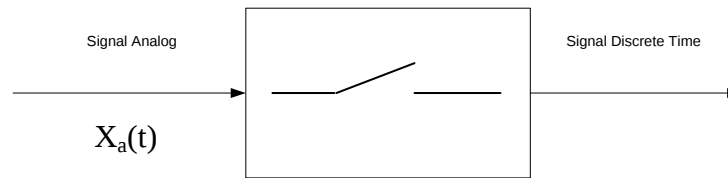
## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengolahan Sinyal**

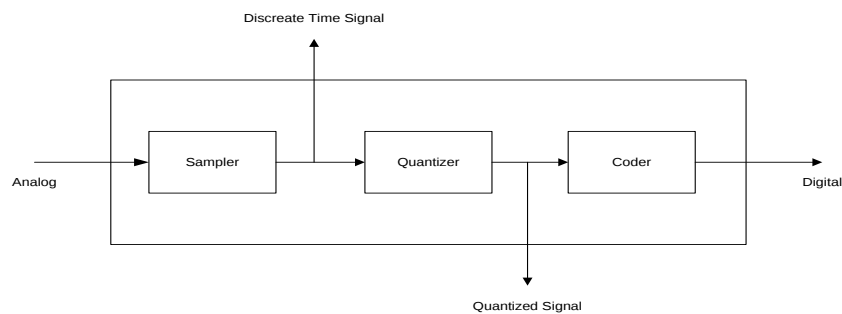
Pengolahan sinyal adalah suatu operasi matematik yang dilakukan terhadap suatu sinyal sehingga diperoleh suatu informasi yang berguna. Dalam hal ini terjadi transformasi (perubahan). Pengolahan sinyal dapat dilakukan secara analog atau digital. Pengolahan sinyal analog memanfaatkan komponen-komponen analog, misalnya dioda, transistor, Op-amp, dan lainnya. Pengolahan sinyal digital menggunakan komponen-komponen digital, register, counter, dekoder, summing, mikroprocessor, mikrokontroler, dan lainnya. Untuk kemudahan pada pengolahan sinyal digital sebagai pemroses digunakan suatu komputer (mikrokontroler) untuk mempresentasikan algoritma atau model matematik. Selain sistem komputer diperlukan perangkat keras lainnya sebagai masukan/keluaran.

Sinyal dapat dibagi menjadi dua jenis yaitu sinyal analog (*continue*) dan sinyal diskrit (*digital*). Pada penelitian ini sinyal yang akan digunakan adalah sinyal diskrit (*digital*), di alam kebanyakan sinyal ditemukan dalam bentuk sinyal analog. Untuk itu, diperlukan pengolahan sinyal dari sinyal analog menjadi sinyal diskrit. Untuk mengolah tersebut dibutuhkan perangkat keras digital. Perubahan sinyal analog menjadi sinyal digital melalui suatu proses yang disebut dengan *sampling*. Konsep dasar prosesnya dapat dilihat pada gambar 2.1 (harlianto , 2007) :



Gambar 2.1 Proses sampling sinyal analog menjadi sinyal digital

Sinyal analog diubah menjadi sinyal digital melalui rangkaian konverter analog to digital converter (ADC). Konverter A/D sudah direalisasikan dalam suatu peranti integrated circuit (IC). Keluaran dari ADC berupa suatu kode biner yang nilainya bersesuaian dengan level kuantisasi dari sinyal analog yang dicuplik pada suatu waktu tertentu. Pada ADC, sinyal analog tersebut diproses melalui tiga proses yaitu *sampling*, *quantizing*, dan *coding*. Prosesnya dapat dilihat pada gambar 2.2 (Harlianto , 2007) :



Gambar 2.2 Proses pengubahan sinyal analog menjadi sinyal digital

Peralatan (*tools*) matematik untuk menganalisis sinyal analog adalah *transformasi laplace*, dan peralatan untuk menganalisis perubahan sinyal analog menjadi sinyal digital adalah transformasi z dan transformasi diskrit fourier.

## 2.2 Pengertian Sinyal

Secara umum, sinyal didefinisikan sebagai suatu besaran fisis yang merupakan fungsi waktu, ruangan, atau beberapa variabel (Harlianto :2007).

Contoh dari sinyal adalah sebagai berikut :

- Tegangan listrik (V) sebagai fungsi waktu.
- Potensial listrik adalah fungsi dari posisi pada suatu ruang 3 dimensi.
- intensitas sebagai fungsi dari kordinat x , y, dan waktu.

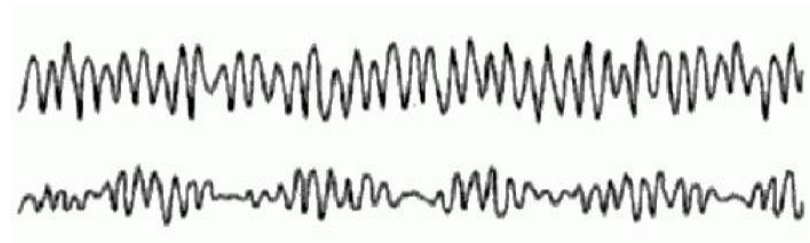
Sinyal dapat dipresentasikan secara matematik, misalnya :

$$\delta_1(t) = 5 t \text{ (linier) } \dots\dots\dots (2.1)$$

$$\delta_2(t) = 20 t^2 + 5 t \text{ (nonlinier/berbanding kuadrat) } \dots\dots\dots (2.2)$$

$$\delta_3(x,y) = 3x + 2xy + 10y \text{ (2 variabel input x dan y) } \dots\dots\dots (2.3)$$

sinyal-sinyal di atas adalah sinyal yang dapat didefinisikan dengan suatu fungsi yang jelas. Dalam beberapa kasus, sinyal tidak dapat dilihat hubungan fungsinya secara nyata dan sangat kompleks. Contohnya sinyal pembicaraan (ucapan manusia) seperti terlihat pada gambar 2.3.



Gambar 2.3 contoh sinyal suara (sumber : Harlianto , 2007)

Suatu sinyal mempunyai beberapa jenis informasi yang dapat diamati , misalnya amplitudo , frekuensi, perbedaan fase dan gangguan akibat *noise*. Untuk mengamati informasi tersebut , dapat digunakan secara langsung dengan peralatan ukur elektronik seperti osciloskop, spektrum analyser. Peralatan tersebut berkerja dengan memanfaatkan model matematik dari sinyal tersebut.

### 2.2.1 Konsep frekuensi

Kebanyakan sinyal di alam ini dalam bentuk analog. Untuk

memperoleh sinyal diskrit dari sinyal analog harus dilakukan proses yang dinamakan sampling (Harlianto : 2007). Secara matematik, proses sampling dinyatakan oleh persamaan 2.4, dan secara diagram blok prosesnya dapat digambarkan oleh gambar 2.4. sinyal analog beragam jenisnya, secara umum (harlianto , 2007) :

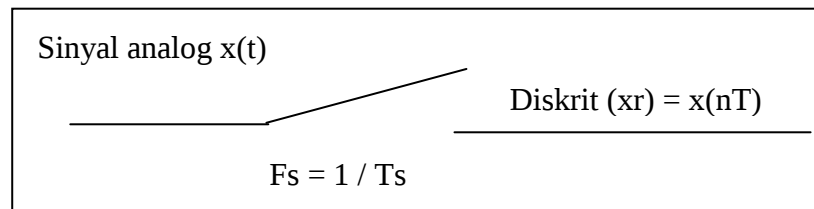
$$X(n) = x_a(nT) = x(t) \mid_{t = Ts}, \text{ untuk } -\infty < n < \infty \text{ (integer) } \dots\dots\dots (2.4)$$

Dimana :

$X(t)$  = sinyal analog

$X(n)$  = sinyal waktu diskrit

$X_a(nT)$  = sinyal analog yang disampling setiap priode  $T_s$



Gambar 2.4 waktu sampling (Harlianto , 2007)

Keterangan ,

$T_s$  = waktu sampling

$F_s = 1/T_s$  , sampling rate (sample/sec)

## 2.3 Algoritma Fast Fourier Transform (FFT)

*Fast fourier transform* (FFT) adalah *transformasi fourier* yang dikembangkan dari algoritma *discrete transfrom fourier* (DFT) . dengan metode FFT, Laju komputasi dari perhitungan transformasi fourier dapat ditingkatkan. Komputasi DFT adalah komputasi yang memerlukan waktu untuk proses *looping* dan memerlukan banyak memori. Dengan menerapkan algoritma FFT, perhitungan DFT dapat dipersingkat, dalam hal ini proses

*looping* dapat direduksi. Dilihat dari metode yang digunakan , FFT dibagi menjadi dua yaitu DIT (*decimation in time*) dan metode DIF (*Decimation in Frequency*), namun keduanya memiliki fungsi yang sama yaitu untuk mentransformasi sinyal menjadi frekuensi dasarnya. *Decimation* adalah proses pembagian sinyal menjadi beberapa bagian yang lebih kecil yang bertujuan untuk memperoleh waktu proses yang lebih cepat. Jika input sinyal pada time domain dari N-points adalah  $x(n)$  ,langkah awal yang dilakukan adalah dengan memisahkan menjadi 2 bagian yang sama ( $N/2$  points).

### 2.3.1 FFT Decimation In Time

*FFT Decimation in time* adalah proses pembagian sinyal menjadi beberapa bagian yang lebih kecil yang bertujuan untuk memperoleh waktu proses yang lebih cepat. FFT diperoleh dengan memodifikasi DFT, modifikasi yang dilakukan adalah dengan cara mengelompokan batas n ganjil dan batas n genap, sehingga N poin DFT menjadi  $(N/2)$  poin. Uraianannya sebagai berikut :

$$X[k] = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad k=0,1,\dots,N-1 \quad \dots\dots\dots (2.5)$$

Adapun langkah – langkah dari algoritma FFT DIT sebagai berikut :

- a** Sinyal input diblok sebanyak N sampel , dimulai dari 0 sampai jumlah sampel mendekati pangkat  $2^N$ . Misal, sinyal akan disampel sebanyak 500 sampel , maka sinyal tersebut akan diblok menjadi 512 ( $2^9$ ).
- b** Tentukan nilai N, yaitu pangkat dari 2 jumlah sampel yang akan diambil, sebagai input sejumlah N-point DFT. Kumpulkan semua sampel dalam buffer sebesar N.

c Rangkaian sinyal input di bit *reserver* dan bagian *imaginer* dibuat nol.  
Contoh : 3 = 011 menjadi 110 = 6.

d Hitung (N/2) dua sampel DFT dari *shuffled inputs*

e Hitung (N/4) empat sampel DFT dari dua sampel DFT

f Hitung (N/2) delapan sampel DFT dari empat sampel DFT

g Lakukan langkah diatas sesuai dengan skema *butterfly* sampai semua sampel dikombinasikan menjadi satu sampel DFT.

h *Buffer* akan terisi dengan hasil dari fourier transfrom.

i Dengan melakukan langkah-langkah diatas, maka spektrum frekuensi x(k) dapat diperoleh, melalui perhitungan :

$$X(k) = \text{DFT}[x(n)] = \sum_{n=-\frac{N_{frame}}{2}}^{\frac{N_{frame}}{2}-1} x_{wc}(n) - e^{\frac{j2\pi nk}{N_{frame}}} \dots\dots\dots (2.6)$$

j N frame sampel DFT di atas merupakan bilangan kompleks yang mempunyai nilai real  $X_r$  dan imajiner  $X_i$ . Untuk memperoleh nilai magnitude  $|X(k)|$  dan phase  $\phi(k)$  dari spektrum tersebut dapat dilakukan dengan perhitungan :

$$|X(k)| = \sqrt{X_r(k)^2 + X_i(k)^2} \dots\dots\dots (2.7)$$

$$\text{Dan, } \Phi(k) = \arctan \frac{X_i(k)}{X_r(k)} \dots\dots\dots (2.8)$$

k Dengan melakukan proses FFT tersebut, maka akan didapat nilai x(k) , dimana masing-masing nilai tersebut merupakan representasi frekuensi dasar dari sinyal masukan.

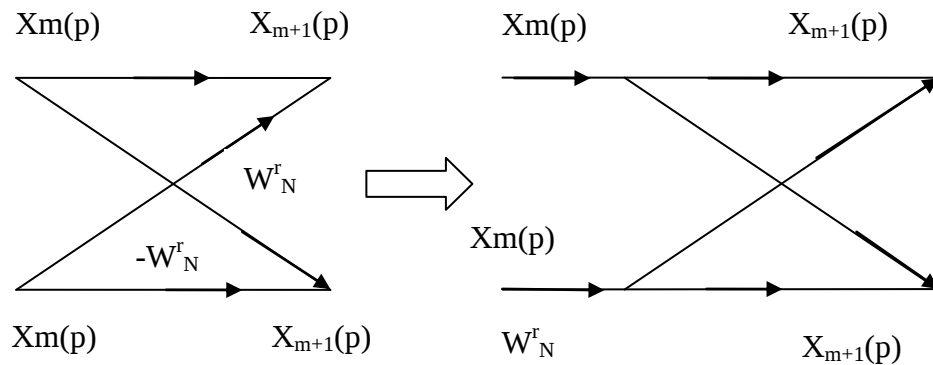
### 2.3.2 FFT Decimation in Frequency

Dengan memodifikasi struktur *butterfly* dari FFT, DIT akan direduksi

proses kalkulasi perkalian dari  $N \log_2 N \rightarrow N/2 \log_2 N$ . Struktur perubahannya dapat dilihat pada gambar 2.5. Bentuk FFT dikenal dengan *FFT Decimation in Frequency (DIF)*.

$$W_N^{r+N/2} = W_N^r \cdot W_N^{N/2} = W_N^r(-1) = -W_N^r \dots\dots\dots (2.9)$$

$$\text{Catatan} = W_N^{N/2} = -1$$



Gambar 2.5 perubahan struktur DFT DIT menjadi DFT DIF

Pada DFT DIF, input tetap seperti asalnya, output disusun atau dikelompokkan ganjil dan genap.

### 2.3.3 Contoh Perhitungan DFT

DFT dilakukan dengan mengimplementasikan sebuah transformasi, dengan panjang vektor  $N$  berdasarkan rumus (Prativi : 2012) :

$$f(u) = \frac{1}{N} + \sum_{x=0}^{x=N-1} f(u) \exp[-j\pi ux/N] \dots\dots\dots (2.10)$$

$$f(u) = \frac{1}{N} + \sum_{N=0}^{x=N-1} f(u) \left( \cos\left(\frac{2\pi ux}{N}\right) - f \sin\left(\frac{2\pi ux}{N}\right) \right) \dots\dots\dots (2.11)$$

Misal diambil data suara hasil kuantitasi sinyal diskrit dengan nilai  $f(x)$  dapat dilihat pada tabel 2.2.

Tabel 2.1 Nilai  $f(x)$  (Prativi : 2012)

|        |        |        |        |
|--------|--------|--------|--------|
| 2      | 3      | 4      | 4      |
| $f(0)$ | $f(1)$ | $f(2)$ | $f(3)$ |

nilai diskrit  $f(x)$  sebanyak 4 data, sehingga dapat ditentukan nilai  $N = 4$  (banyak data), perhitungannya dapat dilihat pada tabel 2.2.

Tabel 2.2 Perhitungan DFT Sinyal Suara (Prativi : 2012)

| N | $f(x)$ | Perhitungan DFT $\sum_{N=0}^{x=N-1}$                                                                                                                                                                                                                                                                                                                   | $f(N)$         |
|---|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| 1 | $f(0)$ | $\frac{1}{4}[2(\cos(2\pi t \cdot 0 \cdot 0/4) - j \sin(2\pi t \cdot 0 \cdot 0/4) + 3(\cos(2\pi t \cdot 0 \cdot 1/4) - j \sin(2\pi t \cdot 0 \cdot 1/4) + 4(\cos(2\pi t \cdot 0 \cdot 2/4) - j \sin(2\pi t \cdot 0 \cdot 2/4) + 4 \cos(2\pi t \cdot 0 \cdot 3/4) - j \sin(2\pi t \cdot 0 \cdot 3/4))]$ $\frac{1}{4}[2(1-0) + 3(1-0) + 4(1-0) + 4(1-0)]$ | 3.25           |
| 2 | $f(1)$ | $\frac{1}{4}[2(1-0) + 3(1-j) + 4(-1) + 4(j)]$                                                                                                                                                                                                                                                                                                          | $-0.5 + 0.25j$ |
| 3 | $f(2)$ | $\frac{1}{4}[2(1-0) + 3(1-2) + 4(1-0) + 4(1-2)]$                                                                                                                                                                                                                                                                                                       | -0.25          |
| 4 | $f(3)$ | $\frac{1}{4}[2(1-0) + 3(j) + 4(-1) + 4(0-j)]$                                                                                                                                                                                                                                                                                                          | $-0.5 - 0.25j$ |

Perhitungan FFT mengimplementasikan pencerminan transformasi ganda hasil DFT dengan hanya menghitung nilai setengahnya data sinyal sehingga perhitungan akan lebih cepat, lalu nilai setengahnya lagi dihitung dengan cara *conjugate* nilai yang telah dihitung dengan DFT. Untuk membagi data sinyal adalah dengan fungsi (Prativi : 2012):

$$b = (N + 1) \text{div } 2 \dots\dots\dots (2.12)$$

Berdasarkan fungsi 2.12 maka didapatkan, jadi perhitungan DFT hanya sampai data ke-3 /  $f(2)$  dan untuk data selanjutnya hanya memakai fungsi *conjugate* dari nilai hasil DFT. Hasil dari nilai DFT dapat dilihat pada tabel 2.3.

Tabel 2.3 Perhitungan FFT dari Hasil DFT (Pratиви , 2012)

| $f(x)$ hasil | Hasil Perhitungan FFT / $F(u)$ |
|--------------|--------------------------------|
| $f(0)$       | 3.25                           |
| $f(1)$       | $-0.5 + 0.25j$                 |
| $f(2)$       | -0.25                          |
| $f(3)$       | $-0.5 - 0.25j$                 |
| $f(4)$       | 3.25                           |

Pemakaian *FFT* karena untuk penghitungan komputasi yang lebih cepat dan mampu mereduksi jumlah perkalian dari  $N^2$  menjadi  $N \log N$  perkalian. *FFT* yang digunakan memakai 512 point dan arena hasil *FFT* simetris, maka keluaran *FFT* tersebut hanya diambil sebanyak 256 data. Hasil dari proses *FFT* akan diperoleh titik-titik sinyal yang simetris sehingga data yang diambil hanya setengah dari data keseluruhan yang selanjutnya akan diambil nilai maksimumnya.

### 2.3 Alat Musik Gitar

Gitar terbagi menjadi dua jenis yaitu gitar akustik dan gitar elektrik. Gitar akustik adalah jenis gitar dimana suara yang dihasilkan berasal dari getaran senar gitar yang dialirkan melalui sadel dan jembatan tempat pengikat senar ke dalam ruang suara. Suara di dalam ruang suara ini akan beresonansi terhadap kayu badan gitar. Jenis dan kualitas kayu serta jenis

senar yang digunakan akan memengaruhi suara yang dihasilkan oleh gitar akustik. Sedangkan gitar elektrik merupakan sejenis gitar yang menggunakan beberapa *pickup* untuk mengubah bunyi atau getaran dari string gitar menjadi arus listrik yang akan dikuatkan kembali dengan menggunakan seperangkat *amplifier* dan *loud speaker*.

Suara gitar listrik dihasilkan dari getaran senar gitar yang mengenai kumparan yang ada di badan gitar yang biasa disebut *pick up*. Terkadang sinyal yang keluar dari *pickup* diubah secara elektronik dengan gitar *effect* sebagai *reverb* ataupun *distorsi*. Disini penulis membahas masalah dengan hanya menggunakan gitar akustik.

Gitar akustik sendiri menghasilkan gelombang yang sama dengan bunyi yaitu gelombang mekanik, dalam bentuk gelombang longitudinal. Gelombang mekanik ini adalah sebuah gelombang yang dalam perambatannya memerlukan medium, yang menyalurkan energi untuk keperluan proses penyaluran sebuah gelombang. Bentuk Gelombang longitudinal adalah gelombang yang memiliki arah getaran yang sama dengan arah rambatan. Artinya arah gerakan medium gelombang sama atau berlawanan arah dengan perambatan gelombang. Gelombang longitudinal mekanis juga disebut sebagai gelombang mampatan atau gelombang kompresi. Karena gitar menghasilkan gelombang, berarti gitar juga memiliki frekuensi tertentu, seperti terlihat pada tabel 2.4 berikut ini :

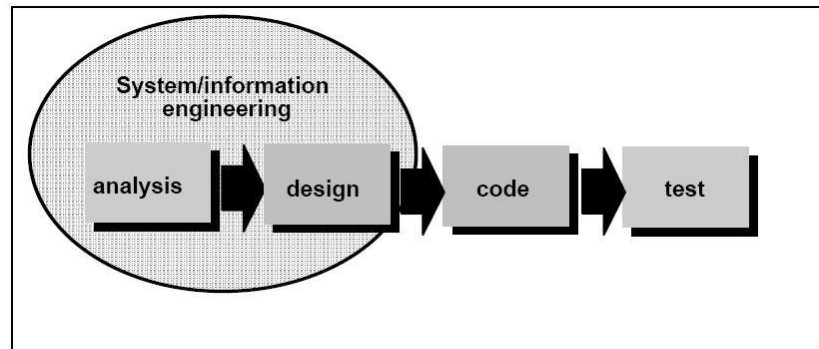
Tabel 2.4 Frekuensi Nada Gitar (Miftahul Huda , 2011)

| Senar   | Notasi Saintis | Notasi Umum | Frekuensi |
|---------|----------------|-------------|-----------|
| Pertama | E5             | e'          | 659,26 Hz |
| Kedua   | B4             | B           | 493,88 Hz |
| Ketiga  | G4             | G           | 392,00 Hz |
| Keempat | D4             | D           | 293,00 Hz |
| Kelima  | A3             | A           | 220,00 Hz |
| Keenam  | E3             | E           | 164,81 Hz |

Frekuensi inilah yang akan digunakan untuk tuning gitar dengan menggunakan algoritma fast fourier transform dan metode Discrete Fourier Transform.

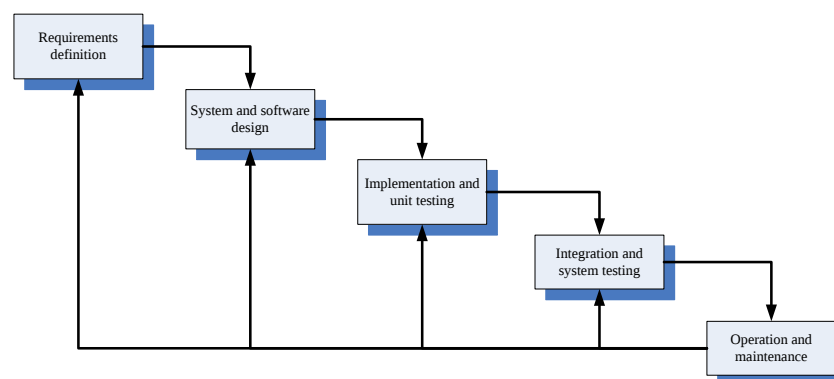
## 2.4 Model Pengembangan *Waterfall*

Model sekuensial linier merupakan salah satu dari metode yang digunakan untuk pengembangan sistem. Sekuensial linier sering disebut juga dengan “siklus kehidupan klasik” atau “model *waterfall*”. Model sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Gambar 2.6 berikut ini menggambarkan model pengembangan sistem sekuensial atau *waterfall* menurut Pressman.



Gambar 2.6 Model *waterfall* (Rosa & M. Shalahuddin, 2009:27)

Sedangkan fase-fase yang terjadi menurut Soummerfile mengembangkan sistem menggunakan model *waterfall* digambarkan pada gambar 2.7.



Gambar 2.7 Fase-fase Model Waterfall (Rosa & M. Shalahuddin, 2009:27)

Masing-masing tahapan yang terdapat pada fase-fase mengembangkan sistem menggunakan model *waterfall* dapat diperjelas berikut ini.

1. Analisis kebutuhan *software*. Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada *software*. Untuk memahami sifat program yang dibangun, analis harus memahami domain informasi, tingkah laku, unjuk kerja, dan *interface* yang diperlukan. Kebutuhan baik untuk sistem maupun *software* didokumentasikan dan dilihat lagi dengan pelanggan.

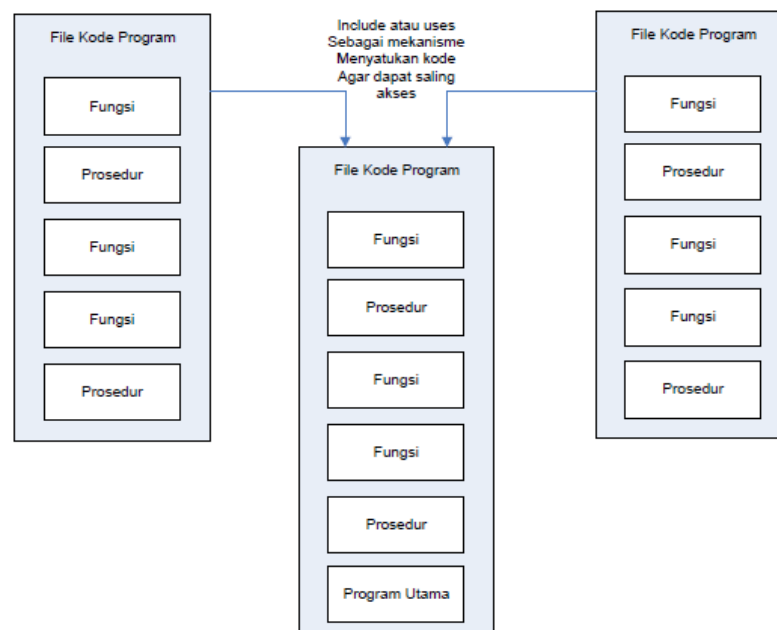
2. Desain. Desain *software* sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda, yaitu struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural. Proses desain menterjemahkan syarat/kebutuhan ke dalam sebuah representasi *software* yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi *software*.
3. Pengkodean. Desain harus diterjemahkan kedalam bentuk mesin yang bisa dibaca. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.
4. Pengujian. Sekali program dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika *internal software*, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa *input* yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.
5. Pemeliharaan. *Software* akan mengalami perubahan setelah disampaikan kepada pelanggan (perkecualian yang mungkin adalah *software* yang dilekatkan). Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena *software* harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat peripheral atau sistem operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan *software*

mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

## 2.5 Pemrograman Terstruktur

Pemrograman terstruktur adalah konsep atau paradigma atau sudut pandang pemrograman yang berbagi-bagi program berdasarkan fungsi-fungsi atau prosedur-prosedur yang dibutuhkan program computer. Modul-modul (pembagian program) biasanya dibuat dengan mengelompokkan fungsi-fungsi dan prosedur-prosedur yang dibutuhkan sebuah proses tertentu.

Fungsi-fungsi dan prosedur-prosedur ditulis secara sekuensial atau terurut dari atas kebawah sesuai dengan kebergantungan antar fungsi atau prosedur (fungsi atau prosedur yang dapat dipakai oleh fungsi atau prosedur dibawahnya harus yang sudah ditulis atau dideklarasikan diatasnya). Berikut ini adalah gambar 2.8 yang merupakan contoh ilustrasi untuk pemrograman terstruktur.



Gambar 2.8 Ilustrasi pemrograman terstruktur

(Rosa & M. Shalahuddin, 2009:62)

## 2.6 Data Flow Diagram (DFD)

Menurut Yourdan dan DeMarco (Rosa & M. Shalahuddin, 2011:64), “*Data Flow Diagram* (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi”. DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi. DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

### 2.6.1 Pembagian Data Flow Diagram (DFD)

Menurut Rosa dan M. Shalahuddin (2001 : 66-67). didalam *Data Flow Diagram* (DFD) terdapat 3 level, yaitu :

#### 1. Diagram Konteks atau diagram level 0

Diagram ini menggambarkan satu lingkaran besar yang dapat mewakili seluruh proses yang terdapat di dalam suatu sistem. Merupakan tingkatan tertinggi dalam DFD dan biasanya diberi nomor 0 (nol). Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran-aliran data

utama menuju dan dari sistem. Diagram ini sama sekali tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan.

## 2. Diagram Level 1

Diagram ini menggambarkan proses-proses yang ada didalam sistem yang akan dibangun. Diagram ini merupakan pemecahan dari diagram konteks yang didalamnya memuat penyimpanan data.

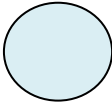
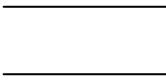
## 3. Diagram Rinci

Diagram rinci merupakan diagram yang menguraikan proses apa yang ada dalam diagram 1 secara lebih jelas. Diagram level 1 dapat diperjelas mulai dari diagram level 2, level 3 dan seterusnya.

### 2.6.2 Komponen Data Flow Diagram

Edward Yourdon dan Tom DeMarco (Rosa & M. Shalahuddin, 2011:65-66) mengemukakan *Data Flow Diagram* memiliki beberapa komponen. Tabel 2.1 berikut ini menampilkan komponen-komponen dari DFD.

Tabel 2.5 Komponen *Data Flow Diagram*

| Notasi                                                                              | Keterangan                                                                                                     |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
|  | Proses atau fungsi atau prosedur, digunakan untuk meng-implementasikan proses yang akan dilakukan dalam sistem |
|  | File atau Basisdata, digunakan untuk menggambarkan basisdata yang terdapat pada sistem                         |

|                                                                                   |                                                                                                                                           |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>Entitas Luar atau masukan atau keluaran atau orang yang memakai / berinteraksi dengan perangkat lunak.</p>                             |
|  | <p>Aliran Data, digunakan untuk menggambarkan aliran data yang terjadi pada sistem, baik aliran data masuk maupun aliran data keluar.</p> |

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Jenis Penelitian**

Jenis-jenis penelitian berbeda-beda dilihat dari berbagai sudut pandang. Berdasarkan tujuan yang ingin dicapai oleh peneliti, penelitian dibagi menjadi dua jenis penelitian yaitu penelitian murni (*pure research*) dan penelitian terapan (*applied research*). Menurut Supranto (Djunaedi, 1993:9), “Penelitian murni adalah penelitian yang bertujuan sebagai pengembangan ilmu pengetahuan berdasarkan atas keinginan hanya untuk mengetahui saja. Sedangkan penelitian terapan adalah penelitian yang hasilnya digunakan untuk membuat suatu keputusan dalam rangka memecahkan persoalan atau menguji hipotesis”.

Berdasarkan definisi kedua jenis penelitian dilihat dari tujuannya, maka penelitian tentang Implementasi algoritma *Fast Fourier Transform (FFT)* Untuk Pengolahan Sinyal Digital Pada *Tuning* Gitar dengan *Open String* ini termasuk dalam penelitian terapan (*applied research*). Penelitian ini menguji manfaat dari teori-teori ilmiah serta mengetahui hubungan empiris dan analisis dalam bidang-bidang tertentu.

Dalam penelitian ini, peneliti berusaha untuk mengetahui tingkat akurasi dari metode *Fast Fourier Transform (FFT)* dalam pengolahan sinyal digital pada senar gitar. Penelitian ini juga dilakukan untuk menghasilkan perangkat lunak yang dapat melakukan tuning gitar.

#### **3.2 Jenis dan Sumber Data**

Jenis data yang dikumpulkan pada penelitian ini adalah data kuantitatif

dalam bentuk sinyal frekuensi pada gitar. Menurut Lofland (1984:47) sebagaimana yang dikutip oleh Lexi J. Moleong bahwa sumber data utama dalam penelitian kualitatif ialah kata-kata dan tindakan selebihnya adalah data tambahan seperti dokumen dan lain-lain. Dimana data hasil penelitian didapatkan melalui dua sumber data, yaitu:

a. Data Primer

Data primer adalah data yang diperoleh atau dikumpulkan oleh peneliti secara langsung dari sumber data utama. Dalam penelitian, untuk mendapatkan data primer peneliti mengumpulkan secara langsung. Adapun data primer itu berupa frekuensi nada gitar yang didapatkan dari percobaan sistem.

b. Data Sekunder

Data sekunder adalah data yang diperoleh atau dikumpulkan oleh peneliti dari berbagai sumber yang telah ada. Dalam penelitian ini data sekunder diperoleh dari berbagai sumber seperti buku, laporan, jurnal, dan lain-lain. Adapun data sekunder dalam penelitian ini seperti frekuensi nada gitar / standar tuning yang digunakan pada gitar oleh miftahul huda dengan judul konversi nada-nada akustik menjadi chord menggunakan pitch class profile.

### 3.3 Objek Penelitian

Obyek penelitian dapat dinyatakan sebagai situasi sosial penelitian yang ingin diketahui apa yang terjadi di dalamnya. Pada obyek penelitian ini, peneliti dapat mengamati secara mendalam aktivitas (*activity*) orang-orang (*actors*) yang ada pada tempat (*place*) tertentu (Sugiyono,

2002). Obyek dari penelitian ini adalah gitar dengan senar enam.

### **3.4 Metode Pengumpulan Data**

Pengumpulan data memiliki beberapa metode. Adapun metode yang digunakan dalam penelitian ini adalah metode studi pustaka. Metode studi pustaka dilakukan dengan cara mempelajari teori-teori literatur dan buku-buku yang berhubungan dengan aplikasi yang akan dibangun dalam penelitian ini seperti algoritma yang digunakan, bahasa pemrograman dan jurnal-jurnal tentang pengolahan sinyal digital.

### **3.5 Metode Pengembangan Sistem**

Pengembangan aplikasi dalam Tugas Akhir ini menggunakan model *waterfall*. Model *waterfall* yang digunakan telah dijelaskan sebelumnya pada bab 2 point 2.6. Adapun penjelasan langkah-langkah yang dilakukan dalam pengembangan sistem ini secara garis besar adalah sebagai berikut.

#### **1) Analisis Kebutuhan**

Aplikasi yang akan dibuat memerlukan masukan, keluaran dan kebutuhan *interface*. Tujuan analisis kebutuhan adalah sebagai batasan dari sistem yang akan dibuat, menentukan kemampuan dan fungsi sistem sesuai dengan kebutuhan pengguna, dan fasilitas-fasilitas yang merupakan nilai tambah yang ada pada sistem yang dibangun. Adapun analisis kebutuhan aplikasi yang akan dibuat adalah sebagai berikut :

##### **a. Kebutuhan data masukan**

Data masukan yang dibutuhkan dalam aplikasi ini adalah bunyi yang dihasilkan oleh senar gitar. Pada penelitian ini , sampel yang digunakan sebanyak 10 sampel untuk setiap nada, sehingga jumlah

sampel yang digunakan adalah 60 sampel untuk 1 gitar.

**b. Kebutuhan data keluaran**

Adapun data keluaran yang dibutuhkan adalah informasi frekuensi senar gitar dan pencocokan nada gitar. Informasi frekuensi senar gitar akan ditampilkan dengan layar digital dan pencocokan nada gitar akan ditampilkan seperti monitoring frekuensi sehingga grafik frekuensi juga akan terlihat, ditambahkan dengan garis petunjuk untuk mempermudah proses *tuning*.

**c. Kebutuhan *interface***

Kebutuhan *interface* pada aplikasi adalah kemudahan dan kenyamanan pengguna saat mengakses aplikasi sesuai dengan permasalahan yang ada.

**2) Perancangan Aplikasi**

Perancangan aplikasi ini adalah tahap konseptualisasi, yaitu suatu tahap yang mengharuskan analisis dalam perancangan sistem (perangkat lunak) untuk berusaha tahu pasti mengenai hal-hal yang menjadi kebutuhan dan harapan pengguna sehingga nanti aplikasi yang dibuat memang dibutuhkan oleh *user* serta memuaskan kebutuhan dan harapannya. Diagram yang digunakan dalam perancangan aplikasi ini adalah *Data Flow Diagram* dikarenakan aplikasi yang akan dibuat merupakan pemrograman terstruktur, untuk lebih jelas dapat dilihat pada bab IV.

**3) Implementasi dan unit pengujian**

Dalam pembuatan aplikasi, tahap ini merupakan tahapan secara nyata

dalam pengerjaan aplikasi. Aplikasi Tuning Gitar akan dimaksimalkan oleh penulis pada tahapan ini. Pembuatan aplikasi dalam penelitian ini menggunakan *software* Matlab.

#### **4) Pengujian sistem**

Pengujian yang dilakukan pada penelitian ini dilakukan dengan melakukan pengujian fungsional dan teknis pada aplikasi yang dibangun, apakah sesuai dengan tujuan dari penelitian ini. yakni apakah perangkat lunak yang dibangun berjalan dengan baik dan benar sehingga dapat mengetahui tingkat akurasi dari algoritma *fast fourier transform* serta mampu melakukan tuning gitar.

#### **5) Penggunaan dan Pemeliharaan**

Setelah aplikasi selesai maka pengguna akan menggunakan aplikasi. Jika terdapat pengembangan fungsional dari aplikasi yang diinginkan oleh pengguna, maka akan dilakukannya pemeliharaan.

### **3.6 Sarana Pendukung**

Dalam pembuatan aplikasi dalam penelitian ini diperlukan sarana pendukung yang berupa beberapa perangkat lunak dan perangkat keras yang membantu menyelesaikan aplikasi. Berikut ini merupakan perangkat lunak dan perangkat keras yang menjadi sarana pendukung penelitian ini.

#### **3.6.1 Perangkat Lunak**

Perangkat lunak yang mendukung aplikasi dalam penelitian ini adalah berupa Sistem Operasi *Windows Seven* (7) 32 bit, Matlab 2008, dan *Microsoft Office Visio 2007* untuk pembuatan diagram alir sistem, *data flow diagram* dan merancang form sistem.

### 3.6.2 Perangkat Keras

Sedangkan perangkat keras yang mendukung dalam penelitian ini adalah 1 unit Laptop Acer dengan spesifikasi monitor VGA atau SVGA (1366 x 768) dan *processor* Intel Core i3, RAM 4 GB, Harddisk 500 GB dan printer *Canon* IP2770.

### 3.7 Metode Pengujian Sistem

Proses pengujian yang dilakukan pada aplikasi yang dibuat menggunakan dua metode pengujian yaitu *white box testing* dan *black box testing*.

#### 1) *White Box Testing*

Dalam pengujian ini, penulis akan meneliti kode-kode program yang ada dan akan menganalisis apakah ada kesalahan atau tidak. Jika terdapat bagian dari kode yang menghasilkan *output* yang tidak sesuai maka penulis akan mengecek satu per satu dan memperbaikinya.

#### 2) *Black Box Testing*

Sedangkan pada pengujian ini, dilakukan dengan mengamati hasil eksekusi antarmuka melalui data uji dan memeriksa fungsional dari aplikasi yang telah dibuat.

### 3.8 Jadwal Penelitian

Jadwal penelitian yang telah dibuat oleh peneliti untuk menyelesaikan penelitian ini dapat dilihat di tabel 3.1 berikut ini.

Tabel 3.1 Jadwal Pelaksanaan Penelitian

| No | Kegiatan                              | Jan.<br>2014 | Feb.<br>2014 | Mar.<br>2014 | Apr.<br>2014 | Mei.<br>2014 | Jun.<br>2014 |
|----|---------------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 1  | Identifikasi masalah                  |              |              |              |              |              |              |
|    | a Latar Belakang Penelitian           |              |              |              |              |              |              |
|    | b Tujuan dan ruang lingkup penelitian |              |              |              |              |              |              |
| 2  | Definisi kebutuhan aplikasi           |              |              |              |              |              |              |
|    | a Pengumpulan data                    |              |              |              |              |              |              |
|    | b Analisis data yang dibutuhkan       |              |              |              |              |              |              |
| 3  | Perancangan aplikasi                  |              |              |              |              |              |              |
| 4  | Pembuatan aplikasi                    |              |              |              |              |              |              |
| 5  | Pengujian aplikasi                    |              |              |              |              |              |              |
| 6  | Analisis Hasil                        |              |              |              |              |              |              |